Lecture Notes in Linear Programming modeling

Dritan Nace

December 19, 2020

Introduction

These lecture notes are exclusively destined to students of UTC. It provides a short introduction of linear programming theory with a special focus on modeling transportation and logistic problems.
Linear programming formulations are popular because the mathematics is nicer, the theory is richer, and the computation simpler for linear problems than for nonlinear ones. In particular several modelers and solvers of high quality are available in the market.

Contents

Chapter 1

Introduction to Linear Programming

1.1 A first example

Let start with an example: Consider a production facility (or an activity) for a manufacturing company. The facility is capable of producing a variety of products that, for simplicity, we shall enumerate as $1, 2, \ldots, n$. These products are manufactured out of certain raw materials. Let us assume that there are m different raw materials, which again we shall simply enumerate as $1, 2, \ldots, m$. The decisions involved in managing/operating this facility are complicated and arise dynamically as market conditions evolve around it. However, to describe a simple, fairly realistic optimization problem, we shall consider a particular snapshot of the dynamic evolution. At this specific point in time, the facility has, for each raw material $j = 1, 2, \ldots, m$, a known amount, say b_j, on hand. Furthermore, each raw material has at this moment in time a known unit market value. We shall denote the unit value of the j^{th} raw material by c_j. In addition, each product is made from known amounts of the various raw materials. That is, producing one unit of product i requires a certain known amount, say a_{ij} units, of raw material j. Also, the i^{th} final product can be sold at the known prevailing market price of p_i euros per unit.

The problem we wish to consider is the one faced by the company's production manager. It is the problem of how to use the raw materials on hand. Let us assume that the manager decides to produce x_i units of the i^{th} product, $i = 1, 2, \ldots, n$. As said above, the revenue associated with the production of one unit of product i is p_i. But there is also a cost of raw materials that must be considered.

The cost of producing one unit of product i is expressed as $\sum_{j=1}^{m} a_{ij} cj$. Therefore, the net revenue associated with the production of one unit is the difference between the revenue and the cost. Since the net revenue plays an important role in our model, we introduce notation for it by setting $f_i = p_i - \sum_{j=1}^{m} a_{ij} cj$, which gives the objectif function:

$$\max \sum_{i=1}^{n} f_i x_i \tag{1.1}$$

Now, let look at the raw materials on hand. Assuming that he decides to produce x_i units of the i^{th} product, $i = 1, 2, \ldots, n$, the manager can't produce more product than he has raw material for. The amount of raw material j consumed by a given production process is a_{ij}, and so he must adhere to the following constraints:

$$\sum_{i=1}^{n} a_{ij} x_i \leq b_j, j = 1, 2, \ldots, m \tag{1.2}$$

Last, requiring nonnegativity:

$$x_i \geq 0, i = 1, 2, \ldots, n \tag{1.3}$$

To summarize, the production manager's job is to determine production values $x_i, i = 1, 2, \ldots, n$, so as to maximize (1.1) subject to the constraints given by (1.2) and (1.3). This optimization problem is an example of a linear programming problem. This particular example is often called the resource allocation problem.

1.2 Linear Optimization

Optimization can be seen as the mathematical branch intended to handle a complex decision problem, involving the selection of values for a number of interrelated variables, by focusing attention on a single objective designed to quantify performance and measure the quality of the decision. The objective is maximizing (or minimizing, depending on the formulation) subject to the constraints that may limit the selection of decision variable values. All dependencies expressed through constraints or objective function are assumed linear.

In a linear program there can be distinguished:

1. Decision variables, that are the unknowns that the user needs to determine.

2. Objective (or optimization) function. This is used to express the objective (minimization or maximization) in terms of the decision variables.

3. Constraints are functions that express the requirements on the relationships between the decision variables.

4. Domain of variables allow to limit the set of decision variables as part of continuousl/integer/positive... numbers.

Standard form of an LP formulation. The general mathematical programming problem can be stated as:

$$\max f(x) \tag{1.4a}$$
$$\textit{subject to}$$
$$h_i(x) = 0, i = 1, 2, \ldots, m \tag{1.4b}$$
$$g_j(x) \leq 0, j = 1, 2, \ldots, p \tag{1.4c}$$
$$x \in S. \tag{1.4d}$$

Where x is an n-dimensional vector of unknowns, $x = (x_1, x_2, \ldots, x_n)$, and f, $h_i, i = 1, 2, \ldots, m$, and $g_j, j = 1, 2, \ldots, p$, are real-valued linear functions of the variables $x_1, x_2, \ldots, x_n$.

The set S is a subset of n-dimensional space. The function f is the objective function of the problem and the equations, inequalities, and set restrictions are constraints. Any particular combination of decision variables is referred to as a solution. A solution is said feasible if it satisfies all constraint requirements.

In this course we will focus on a specific formulation called *standard formulation* which is specified below:

$$\max c_1x_1 + c_2x_2 + \cdots + c_nx_n \tag{1.5a}$$
$$\textit{subject to}$$
$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \tag{1.5b}$$
$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \tag{1.5c}$$
$$\cdots \tag{1.5d}$$
$$a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \tag{1.5e}$$
$$x_i \geq 0, \forall i \in \{1, \ldots, n\}. \tag{1.5f}$$

Note that each of the constraints should be arranged such that all terms with decision variables be on the left of the constraint realation, and the only part that appears on the right is the constant term.

Converting various forms of LP to the standard form can be done as follows:

1. (Slack variables). Consider the problem as in the standard form but constraints are typically: $a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \leq b_i$. These constraints can be rewritten as equalities by adding a new variable (slack variable) to the left hand side.

2. (Surplus variables). If the above linear inequalities are reversed: $a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \geq b_i$. These constraints can be rewritten as equalities by substracting a new variable (surplus variable) to the left hand side.

3. (Free variable). If $x_i \geq 0$ is not present and hence x_i is free to take on either positive or negative values. We then write $x_i = u_i - v_i$, where $u_i \geq 0$ and $v_i \geq 0$. An alternative for converting to standard form when x_i is unconstrained in sign is to eliminate x_i together with one of the constraint equations.

Example. Consider the following problem:

$$\max x_1 + 3x_2 + 4x_3 \tag{1.6a}$$
$$\textit{subject to}$$
$$x_1 + 2x_2 + x_3 = 5 \tag{1.6b}$$
$$2x_1 + 3x_2 + x_3 = 6 \tag{1.6c}$$
$$x_2 \geq 0, x_3 \geq 0. \tag{1.6d}$$

Considering $x_1 = 5 - 2x_2 - x_3$ the above can be rewritten as:

$$\max x_2 + 3x_3 \tag{1.7a}$$
$$\textit{subject to}$$
$$x_2 + x_3 = 4 \tag{1.7b}$$
$$x_2 \geq 0, x_3 \geq 0. \tag{1.7c}$$

1.3 Linear Program modeling

Although in the majority of cases modelling a LP problem is straightforward, in some others it may be needed to introduce new variables and/or to rearrange the constraints. We will cite a few examples below:
In some situations expressing the objective function may need using maxmin or minmax functions. Let be given variables $x_1, x_2, \ldots, x_n$, and the objective function consists in maximizing the minimum of them: $\max\{\min\{x_1, x_2, \ldots, x_n\}\}$. This leads to a maxmin function which cannot be expressed straightly by linear programming. A simple way to deal with it is to introduce a new variable y which is a lower bound of x_j, and state the problem as follows:

$$\max y \tag{1.8a}$$
$$\textit{subject to}$$
$$x_j \geq y, j = 1, \ldots, n \tag{1.8b}$$
$$\ldots$$

In some other situations some *multiplicative* functions may be found. A simple example is $\min\{1/x\}$ which may be simply expressed as $\max\{x\}$. In some other situations we may have bilinear terms ax where both a and x are decision variables. Although in the general case such a formulation cannot be linearized, it can be easily handled when one of them is binary and the other is upperly bounded. Let suppose that x is a binary variable and $a \in \{0, .., M\}$. Than let first introduce a new variable $y = ax$. This new variable (y) is used instead of expression ax wherever it appears in the model. Now one needs to add new constraints to make $y(\geq 0)$ to behave properly, that is,

$$y \leq a \tag{1.9a}$$

$$y \leq Mx \tag{1.9b}$$

$$y \geq a + (x-1)M \tag{1.9c}$$

where M is a large value constant. The case when both are binary variables is handled similarly.

Different situations may be met when modeling constraints as well. For instance, the so-called *capacity constraints* usually give rise to upper bound constraints; In the same line, *demand constraints* will usually give rise to lower bound constraints. Another type of frequently met constraint is the socalled *mass balance* one, which expresses the requirement of conservation of mass. An obvious example of this constraint is found in flow problems when expressing that the amount of flow entering in some transit node is equal to the amount of flow going out.
Another type of constraints are the proportion constraints where one decision variable needs to represent a fixed part of a given sum. For instance, let a, b and c give all, and a is required to be exactly 30% of all. Then, it can be expressed as: $a = 0.3(a + b + c)$.

1.4 Examples of Linear Programming problems

1.4.1 A simplified production problem

Determine the quantities to be produced such that all the production constraints are satisfied and the profit is maximized. We suppose that two products A and B can be produced, each of them passing through cutting and packing stages: Necessary time to produce 1 unity of A is 2 hours for cutting and 3 hours for packing. Necessary time to produce 1 unity of B is 2 hours for cutting and 1 hour for packing. Availability in working hours is 200 hours for cutting and 100 hours for packing. The unitary profits for A and B are respectively 20 and 10 euros.
This problem falls in the category of *Resource Allocation problems* studied in Section 1.1. Writing the LP formulation is left to the reader.

1.4.2 The knapsack problem

The knapsack problem (KP) has a significant place in the study of integer programming models with binary variables. In the knapsack problem, one needs to pack a set of items (I), with given values (v) and sizes (w) (such as weights or volumes), into a container with a maximum capacity (C). If the total size of the items exceeds the capacity, we can't pack them all. In that case, the problem is to choose a subset of the items of maximum total value that will fit in the container. The decision variables are represented by binary variables x_i taking value 1 if item i is selected and packed into the container and 0 otherwise. The objective function is max $\sum_{i \in I} v_i x_i$ such that $\sum_{i \in I} w_i x_i \leq C$.

1.4.3 The diet problem

We assume that there are available at the market n different foods and that the j^{th} food sells at a price c_j per unit. In addition there are m basic nutritional ingredients and, to achieve a balanced diet, each individual must receive at least b_i units of the i^{th} nutrient per day. Finally, we assume that each unit of food j contains a_{ij} units of the i^{th} nutrient.

If we denote by x_j the number of units of food j in the diet, the problem then is to select the x_j's to minimize the total cost: $c_1x_1 + c_2x_2 + \cdots + c_nx_n$ subject to the nutritional constraints: $a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \geq b_i, i = 1, \ldots, m$, and the non-negativity constraints: $x_j \geq 0$ for all j.

1.4.4 The art gallery problem

The art gallery problem is a well-studied optimization problem. It originates from a real-world problem of guarding an art gallery with the minimum number of guards who together can observe the whole gallery. In the geometric version of the problem, the layout of the art gallery is represented by a simple polygon and each guard is represented by a point in the polygon. In the graph theory version the problem is represented as the vertex cover of an undirected graph problem. A vertex cover of an undirected graph $G = (V, E)$ is a subset of its vertices such that for every edge (u, v) of the graph, either u or v is in the vertex cover. Although the name is Vertex Cover, the set covers all edges of the given graph. Given an undirected graph, the vertex cover problem is to find minimum size vertex cover. If we denote by x_i the binary variable indicating if vertex i is in the Vertex Cover or not, the problem then is to select the x_i's to minimize: $x_1 + x_2 + \cdots + x_n$ subject to the covering constraints: $x_i + x_j \geq 1, \forall (i, j) \in E$ and the binary constraints: $x_i \in \{0, 1\}$ for all vertex i.

1.4.5 A warehousing problem

Consider the problem of operating a warehouse, by buying and selling the stock of a certain commodity, in order to maximize profit over a certain length of time, say n months. The warehouse has a fixed capacity C, and there is a cost r per unit for holding stock for one period. The selling price, p_i, of the commodity is known to fluctuate over a number of time periods n indexed by i. In any period i the price holding for purchase is f_i. The warehouse is originally empty and is required to be empty at the end of the last period.

To formulate this problem, variables are introduced for each time period. In particular, let z_i denote the level of stock in the warehouse at the beginning of period i. Let x_i denote the amount bought during period i, and let y_i denote the amount sold during period i. If there are n periods, the problem may be written as:

$$\min \sum_{i=1}^{n} f_i x_i - \sum_{i=1}^{n} p_i y_i + \sum_{i=1}^{n} z_i r \tag{1.10a}$$

subject to

$$x_i + z_i - y_i = z_{i+1} \quad 1 \leq i \leq n \tag{1.10b}$$

$$z_i \leq C \quad 1 \leq i \leq n \tag{1.10c}$$

$$z_1 = z_{n+1} = 0 \tag{1.10d}$$

$$x, y, z \geq 0. \tag{1.10e}$$

1.4.6 Exercises

1. Convert the following problem to standard form:

$$\max x + 2y + 3z \tag{1.11a}$$
$$subject\ to$$
$$6 \leq x + y + z \leq 15, \tag{1.11b}$$
$$x \geq 2, y \geq 0, z \geq 0 \tag{1.11c}$$

2. Convert the following problem to standard form and solve:

$$\max x_1 + 4x_2 + x_3 \tag{1.12a}$$
$$subject\ to$$
$$2x_1 - 2x_2 + x_3 = 4 \tag{1.12b}$$
$$x_1 - x_3 = 1 \tag{1.12c}$$
$$x_2 \geq 0, x_3 \geq 0. \tag{1.12d}$$

3. Convert the following problem to a linear program in standard form:

$$\min |x| + y + z \tag{1.13a}$$
$$subject\ to$$
$$x + y \leq 1 \tag{1.13b}$$
$$2x + z = 3 \tag{1.13c}$$
$$y, z \geq 0 \tag{1.13d}$$

4. A class of piece-wise linear functions can be represented as $f(x) = \max(c_1^T x + d_1, c_2^T x + d_2, ..., c_p^T x + d_p)$. For such a function f , consider the problem minimize $f(x)$ subject to $Ax = b, x \geq 0$. Show how to convert this problem to a linear programming problem

5. A manufacturer wishes to produce to produce 1000 lb of an alloy that is, by weight, 30% metal A and 70% metal B. Five alloys are available at various prices as indicated below:

Alloy	1	2	3	4	5
%A	10	25	50	75	95
% B	90	75	50	25	5
Price/lb	5	4	3	2	1.50

The desired alloy will be produced by combining some of the other alloys. The manufacturer wishes to find the amounts of the various alloys needed and to determine the least expensive combination. Formulate this problem as a linear program.

6. A small firm specializes in making five types of spare automobile parts. Each part is first cast from iron in the casting shop and then sent to the finishing shop where holes are drilled, surfaces are turned, and edges are ground. The required worker-hours (per 100 units) for each of the parts of the two shops are shown below:

Part	1	2	3	4	5
Casting	2	1	3	3	1
Finishing	3	2	2	1	1

The profits from the parts are 30, 20, 40, 25, and 10 (per 100 units), respectively. The capacities of the casting and finishing shops over the next month are 700 and 1000 worker-hours, respectively. Formulate the problem of determining the quantities of each spare part to be made during the month so as to maximize profit.

7. A small computer manufacturing company forecasts the demand over the next n months to be d_i, $i = 1, 2, \ldots, n$. In any month it can produce at most r units, using regular production, at a cost of b dollars per unit. By using overtime, it can produce additional units at c dollars per unit, where $c > b$. The firm can store units from month to month at a cost of s dollars per unit per month. Formulate the problem of determining the production schedule that minimizes the cost.

Chapter 2

Logistic, transportation and scheduling problems modelling

In today's global economies, transportation is a key facilitator of trade, and hence an important factor in rising prosperity and welfare. Natural resources are scarce and not evenly distributed in terms of type and geographical location in the world. Transportation and logistic chains enable the distribution of materials, food and products from the locations where they are extracted, harvested or produced to people's homes and nearby stores. At the same time, current logistics systems are fundamentally unsustainable, due to the emission of CO2, congestion, noise and the high price that has to be paid in terms of infrastructural load. Hence, there are problems that are necessary to solve in practice and transport and logistics problems belong among them. The correct optimization of such problems enables us to minimize the cost and time.

Another group of interesting problems comes from scheduling. Scheduling is concerned with the allocation of scarce resources to activities with the objective of optimizing one or more objectives. Depending on the situation, resources may be machines in an assembly plant, CPU or memory in a computer system, runways at an airport, etc. With respect to above, activities may be various operations in a manufacturing process, execution of a computer program, landings and take-offs at an airport, and so on. There also may be different criterion to optimize. One objective may be the minimization of the makespan, while another objective may be the minimization of the number of late jobs.

The focus of this chapter is in introducing some typical optimization problems in each of above areas together with their linear formulations.

2.1 Logistic problems

Among the logistic problems, the facility Location Problem is perhaps one of the most important topics in industrial engineering and operations research. It aims to take strategic decisions for distribution of goods, services, and information. There are plenty of applications for hub location problem, such as Transportation Management, Urban Management, locating service centers, Instrumentation Engineering, design of sensor networks, Computer Engineering, design of computer networks, Communication Networks Design, Power Engineering, localization of repair centers, and Design of Manufacturing Systems. Today, we note that the markets are becoming larger, but often with a more dispersed clientele, deliveries easier thanks to the opening of borders but more competitive. In the problems we are presenting, the general objective is to determine the optimal location of a number of facilities on a set of possible sites, so as to minimize the number of installations to cover all demands, or maximize the demand covered with a fixed number of installations, or minimize costs when delivery issues are also concerned (the facility location-allocation), it depends on the application on hand. Then, the questions to be answered are: How many installations? Where to place them? How to assign requests to sites?

2.1.1 Single installation

Consider first a very simple case: the location of a single installation. Data is composed of m possible sites, n entities (customers, suppliers, etc.) to be served, and a simple load measurement L_j for each entity j, to be defined. Depending on the activity, this may be a tonnage of goods entering or leaving the site, a number of weekly trips, etc. Also, distances d_{ij} between any site i and any entity j are known. The objective is to choose a site i^* minimizing a score estimating the work of moving loads.

The solution method is simply calculating for any site i of a load-distance score $ld(i)$, and choosing the site with the minimum score.

$ld(i) = \sum_{j=1..n} L_j d_{ij}$

A special case of above is the "Center of gravity" problem. This is a variant of continuous localization which consists of determining the center of gravity of loads, i.e. the geographical location (x^*, y^*), e.g. longitude and latitude, is to be determined, so we only consider the entities to be served, (whose coordinates $(x_i y_i)$ are known), and not the sites.

Question: Give the formulas for calculating x^* and y^*.

2.1.2 Full coverage

The problem of full coverage raises when one seeks to cover all clients/requests in a given area. The problem may be expressed as follows.
Data:

- n potential sites (indexed by i),
- m demand points (indexed by j),
- distances d_{ij} between any site i and any entity j.
- D_c coverage distance beyond which a customer cannot be served,
- constants (0,1) denoted by a_{ij} indicating whether the point of demand j is covered by the site i, that is $a_{ij} = 1 \simeq d_{ij} \leq D_c$.
- c_i cost of opening a site i,

Objective: calculate the sites to be opened to cover all the requests, with a minimum total cost.

Binary linear programming formulation:

$$\min \sum_{i=1}^{n} c_i x_i \tag{2.1a}$$

subject to

$$\sum_{i=1}^{n} a_{i,j} x_i \geq 1, \quad \forall j \in \{1, \ldots, m\} \tag{2.1b}$$

$$x_i \in \{0, 1\}, \quad \forall i \in \{1, \ldots, n\}. \tag{2.1c}$$

2.1.3 Maximum coverage

In contrast to the precedent problem, we give up total coverage and seek for maximizing the demand covered with a limited number of sites to open. Parameters and additional variables are:

- p: maximum number of sites to open,
- d_j: request/demand of point j,
- z_j: binary variable indicating whether demand j is covered.

Formulation of the problem as a binary linear program follows:

$$\max \sum_{j=1}^{m} d_j z_j \tag{2.2a}$$

subject to

$$z_j \leq \sum_{i=1}^{n} a_{i,j} x_i \quad \forall j \in \{1, \ldots, m\} \tag{2.2b}$$

$$\sum_{i=1}^{n} x_i \leq p, \tag{2.2c}$$

$$x_i \in \{0, 1\}, \quad \forall i \in \{1, \ldots, n\} \tag{2.2d}$$

$$z_j \in \{0, 1\}, \quad \forall j \in \{1, \ldots, m\}. \tag{2.2e}$$

2.1.4 P-center problem

The p-center problem is a relatively well known facility location problem that involves locating p identical facilities on a network to minimize the maximum distance between demand nodes and their closest facilities. The focus of the problem is on the minimization of the worst case service time.

Parameters and additional variables are:

- n: a set of a number of nodes to serve,
- a set of n sites, of which only p sites can open,
- Distance matrix d_{ij} between sites and nodes,
- y_{ij}: a binary variable taking 1 if site i is covering demand j,

- z: is the maximum covering distance.

The objective is to place the sites to minimize the maximum intervention distance. An example is placement of fire stations, etc.

$$\min z \tag{2.3a}$$

$$\textit{subject to}$$

$$\sum_{i=1}^{n} y_{ij} = 1, \quad \forall j \in \{1, \ldots, m\} \tag{2.3b}$$

$$\sum_{i} x_i \leq p, \tag{2.3c}$$

$$\sum_{i=1}^{n} d_{ij} y_{ij} \leq z, \quad \forall j \in \{1, \ldots, m\} \tag{2.3d}$$

$$y_{ij} \leq x_i, \quad \forall i \in \{1, \ldots, n\}, \forall j \in \{1, \ldots, m\} \tag{2.3e}$$

$$x_i \in \{0, 1\}, \quad \forall i \in \{1, \ldots, n\} \tag{2.3f}$$

$$y_{ij} \in \{0, 1\}, \quad \forall i \in \{1, \ldots, n\}, \forall j \in \{1, \ldots, m\}. \tag{2.3g}$$

2.1.5 P-median problem

In contrast to the p-center problem, the p-median one looks for minimizing the average intervention distance. An example is the placement of warehouses to minimize the average travel time to deliver a customer. So, the p-median problem seeks to minimize the average intervention distance, possibly weighted by the demands of the nodes denoted with q_j below. Linear programming formulation follows:

$$\min \sum_{j=1}^{m} q_j \sum_{i} d_{ij} y_{ij} \tag{2.4a}$$

$$\textit{subject to}$$

$$\sum_{i=1}^{n} y_{ij} = 1, \quad \forall j \in \{1, \ldots, m\} \tag{2.4b}$$

$$\sum_{i=1}^{n} x_i \leq p, \tag{2.4c}$$

$$y_{i,j} \leq x_i, \quad \forall i \in \{1, \ldots, n\}, \forall j \in \{1, \ldots, m\} \tag{2.4d}$$

$$x_i \in \{0, 1\}, \quad \forall i \in \{1, \ldots, n\} \tag{2.4e}$$

$$y_{i,j} \in \{0, 1\}, \quad \forall i \in \{1, \ldots, n\}, \forall j \in \{1, \ldots, m\}. \tag{2.4f}$$

2.1.6 Warehouse location

This problem arises when one seeks to minimize the total cost including the fixed cost of opening and operating the sites and the cost of transportation. It is then necessary to decide which sites to open and what quantities to deliver from any site to any customer. This classic problem is called the warehouse location problem. We denote with f_i the opening cost, d_j te demand corresponding to certain period, and with c_{ij} the delivery cost from i to j. Variables y_{ij} denote the quantity of demand j covered by site i and M gives a sufficiently large constant.

Linear programming formulation follows:

$$\min \sum_{i=1}^{n} f_i x_i + \sum_{j=1}^{m} \sum_{i=1}^{n} c_{ij} y_{ij} \tag{2.5a}$$

subject to

$$\sum_{i=1}^{n} y_{ij} = d_j, \quad \forall j \in \{1, \ldots, m\} \tag{2.5b}$$

$$\sum_{j=1}^{m} y_{i,j} \leq M x_i \quad \forall i \in \{1, \ldots, n\}, \tag{2.5c}$$

$$x_i \in \{0, 1\}, \quad \forall i \in \{1, \ldots, n\} \tag{2.5d}$$

$$y_{ij} \geq 0, \quad \forall i \in \{1, \ldots, n\}, \forall j \in \{1, \ldots, m\}. \tag{2.5e}$$

2.2 Transportation problems

There are various transport issues between sources with given availabilities and destinations with given demands to be satisfied. Links in the network have costs that should be taken in account when writing down the cost (objective) function for problems like:

- transport problem,
- assignment problem,
- transshipment problem,
- optimal path problems,
- vehicle routing problems.

In some other cases, links have capacities:

- maximum flow problem,
- minimum cost flow problem,
- multi-flow problem.

2.2.1 The transportation problem

Quantities $a_1, a_2, ..., a_n$, respectively, of a certain product are to be shipped from each of n locations and received in amounts $b_1, b_2, \ldots, b_m$, respectively, at each of m destinations. A shipping cost c_{ij} is associated with the shipping of a unit of product from origin i to destination j. It is desired to determine the amounts x_{ij} to be shipped between each origin-destination pair $i = 1, 2, \ldots, n; j = 1, 2, \ldots, m$; so as to satisfy the shipping requirements and minimize the total cost of transportation. The problem is representable graphically by a bipartite graph.

$$\min \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} x_{ij} \tag{2.6a}$$

$$\textit{subject to}$$

$$\sum_{j=1}^{m} x_{ij} \leq a_i, \quad i = 1, 2, \ldots, n \tag{2.6b}$$

$$\sum_{i=1}^{n} x_{ij} = b_j, \quad j = 1, 2, \ldots, m \tag{2.6c}$$

$$x_{ij} \geq 0, \quad i = 1, 2, ..., n; j = 1, 2, \ldots, m. \tag{2.6d}$$

The assignment problem is a transportation problem with availability and requests fixed at 1 and $n = m$.

In the transsshipment problem, the graph has infinite capacity arcs but is no longer bipartite, that is, some intermediate nodes/destinations are present in the graph.

2.2.2 The travelling salesman problem

The travelling salesman problem, known as TSP problem, one of the most famous in the group of vehicle routing problems and more largely in combinatorial optimization. It consists in computing, for a given list of cities and the distances between each pair of cities, the shortest possible route that visits each city exactly once and returns to the origin city. The travelling salesman problem was mathematically formulated in the 1800s by the Irish mathematician W.R. Hamilton, which problem is known in graph theory as the Hamiltonian cycle one. Let $G = (V, E)$ be the graph composed of n nodes (V) representing cities and arcs (E) representing direct links between cities weighted by their distance (d_{ij}). Let x_{ij} represent the decision variables taking 1 if arc (ij) is chosen to be in the circuit and 0 otherwise. The problem may be formulated as follows:

$$\min \sum_{i,j} d_{ij} x_{ij} \tag{2.7a}$$

$$\textit{subject to}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \forall i = 1, 2, \ldots, n \tag{2.7b}$$

$$\sum_{i=1}^{n} x_{ij} = 1, \forall j = 1, 2, \ldots, n \tag{2.7c}$$

$$\sum_{i \in S, j \in V \setminus S} x_{ij} \geq 1, \forall S \subset V \tag{2.7d}$$

$$x_{ij} \in \{0, 1\}, \forall i = 1, 2, \ldots, n, j = 1, 2, \ldots, n \tag{2.7e}$$

Clearly constraints (2.7d) are very numerous and cannot be all put explicitly. Then, we start with a subset of such constraints and solve the problem. The obtained solution x is necessarily composed of circuits and all nodes are visited. If the solution corresponds to a single circuit, then the solution is optimal, otherwise one can build a constraint of type (2.7d) by putting in a set S all nodes contained in a sub-tour and next add it in the formulation and solve again the augmented problem[1].

2.2.3 The maximal flow problem

Consider a capacitated network $N = (V, E, C)$ with n nodes (V) in which two special nodes, called the source s and the sink t, are distinguished. All other nodes must satisfy the strict flow conservation requirement; that is, the net flow into these nodes must be zero. The outflow f of the source will equal the inflow of the sink as a consequence of the conservation at all other nodes. There is also a defined a capacity vector C. An arc connecting node i to node j is denoted with $(i, j) \in E$. The flow passing on each arc should satisfy the capacity constraint, that is sum of flows passing through arc (i, j) should be less or equal to C_{ij}. A set of arc flows satisfying these conditions is said to be a flow in the network of value f. $U^+(i)$ and $U^-(i)$ denote respectively the out-neighbourhood and the in-neighbourhood of node i. The maximal flow problem is that of determining the maximal flow that can be established in such a network.

$$\max f \tag{2.8a}$$

$$subject\ to$$

$$\sum_{j \in U^+(s)} x_{sj} - \sum_{j \in U^-(s)} x_{js} - f = 0, \tag{2.8b}$$

$$\sum_{j \in U^+(i)} x_{ij} - \sum_{j \in U^-(i)} x_{ji} = 0,\ i = \{1, \ldots, n\} \setminus \{s, t\} \tag{2.8c}$$

$$x_{ij} \leq C_{ij},\ (i, j) \in E, \tag{2.8d}$$

$$x_{ij} \geq 0,\ (i, j) \in E. \tag{2.8e}$$

2.2.4 The minimum cost flow problem

The minimum cost flow problem is that of determining a flow of a given value f of minimal cost. The cost is represented by vector w. Then, the problem can be formulated as the maximum flow problem instead that f is now a parameter and the objective function becomes:

$$\min \sum_{(i,j) \in E} w_{i,j} x_{i,j}$$

[1] Other formulations are also possible. Another option to avoid sub-tours is to add some variables and constraints. Then, instead of constraints (2.7d), we add new variables u_i (one per node) and constraints: $u_i - u_j \leq n - 1; 2 \leq i \neq j \leq n, u_i \in \mathcal{N}$ It can be shown that this constraint avoids having sub-tours in the obtained solution.

2.2.5 The multi-flow problem

The related multi-flow problems are like the flow one but several commodities, called $d, d \in \mathcal{D}$, are sharing the same network. One commodity is determined by its origin (or source) and destination (or sink) nodes $s(d)$ and $t(d)$, and its achieved value f_d. The minimum cost multi-commodity flow problem, where f_d is a given parameter, is formulated as:

$$\min \sum_{(i,j)\in E} w_{i,)} \sum_{d} x^d_{i,j} \tag{2.9a}$$

subject to

$$\sum_{j\in U^+(sd))} x^d_{s(d),j} - \sum_{j\in U^-(s(d))} x^d_{j,s(d)} - f_d = 0, \quad \forall d \in \mathcal{D} \tag{2.9b}$$

$$\sum_{j\in U^+(i)} x^d_{ij} - \sum_{j\in U^-(i)} x^d_{ji} = 0, \quad \forall d \in \mathcal{D}, \forall i \in V \setminus \{s(d), t(d)\} \tag{2.9c}$$

$$\sum_{d\in\mathcal{D}} x^d_{ij} \leq C_{ij}, \quad \forall (i,j) \in E \tag{2.9d}$$

$$x^d_{ij} \geq 0, \quad \forall d \in \mathcal{D}, \forall (i,j) \in E. \tag{2.9e}$$

Exercises

1. The minimum cost maximum flow problem is that of determining among the maximal flows that can be established in such a network, the one of minimal cost. How one can solve this problem?
2. Use the minimum cost flow problem formulation given above to formulate the minimum shortest path problem.

2.3 Scheduling problems

The study of scheduling dates back to 1950s. Researchers in operations research, industrial engineering,and management were faced with the problem of managing various activities occurring in a workshop. Next, beginning in the late 1960s, computer scientists also encountered scheduling problems in the development of operating systems. Back in those days, computational resources (such as CPU, memory and I/O devices) were scarce. Efficient utilization of these scare resources can lower the cost of executing computer programs. This provided an economic reason for the study of scheduling.

Today, the community is interested in different problems like single machine problems, flow-shop and job-shop scheduling problems, the resource-constrained project scheduling problem, etc. In general, these problems are tackled by specific algorithms, but more recently other methods as constraint programming and mathematical (linear) programming are also studied. In the following we will look at two well-known problems that are the *one machine problem* and *the number of delayed tasks.*

Generally speaking, problem formalization of scheduling problems is concerned with data including tasks/jobs (ready time, processing time, due time), number of machines, precedence constraints, preemption, etc. The notation used are as follows:

- Parameters (per task i): r_i (ready time), p_i (processing time), d_i (due time);
- Decision variables: t_i (start date of execution of task i), x_{ij} (execution of task i at order j), c_i (completion execution time of task i).
- Precedence/conjunctive constraints such that task i before task j are expressed as $t_i + p_i \leq t_j$;
- Disjunctive constraints may be expressed as: $t_i + p_i \leq t_j$ or $t_j + p_j \leq t_i$; $t_i + p_i \leq t_j + M(1-b); t_j + p_j \leq t_i + Mb; b \in \{0, 1\}$, where M is a sufficiently large value.
- Objective function is generally concerned with minimization of completion time (makespan), total lateness, number of delayed tasks, etc.

2.3.1 The one-machine problem

We consider the n-job one-machine scheduling problem with ready time r_i, processing time p_i, and due time d_i for each job i. Preemption is not allowed, and precedence constraints among jobs are not considered. Decision variables: x_{ij} (boolean variables): task i executed at the j^{th} order; t_j (≤ 0) start time for task placed at order j;

LP formulation of one-machine problem:

$$\min t_n + \sum_i x_{in} p_i \tag{2.10a}$$

$$subject\ \ to$$

$$\sum_{j=1}^{n} x_{ij} = 1, \quad \forall i = 1, 2, \ldots, n \tag{2.10b}$$

$$\sum_{i=1}^{n} x_{ij} = 1, \quad \forall j = 1, 2, \ldots, n \tag{2.10c}$$

$$t_j \geq \sum_{i=1}^{n} x_{ij} r_i, \quad \forall j = 1, 2, \ldots, n \tag{2.10d}$$

$$t_j + \sum_{i=1}^{n} x_{ij} p_i \leq t_{j+1}, \quad \forall j = 1, 2, \ldots, n \tag{2.10e}$$

$$t_j + \sum_{i=1}^{n} x_{ij} p_i \leq \sum_{i=1}^{n} x_{ij} d_i, \quad \forall j = 1, 2, \ldots, n \tag{2.10f}$$

$$x_{ij} \in \{0, 1\}, \quad \forall i = 1, 2, \ldots, n, j = 1, 2, \ldots, n \tag{2.10g}$$

2.3.2 Number of delayed tasks problem

In contrast with the one-machine problem, the *the number of delayed tasks* problem concerns a set $I = 1, 2, ..n$ of tasks of duration 1 and deadlines d_i, a partial order $<$ denoted with $Prec$ on I. The objective is to achieve a scheduling of these tasks on one machine satisfying the partial order and such that the number of delayed tasks is the smallest possible.

LP formulation:

$$\min \sum_j y_j \tag{2.11a}$$

$$subject \ \ to$$

$$t_i \leq t_j \quad \forall (i \leq j) \in Prec, \tag{2.11b}$$

$$t_j + 1 \leq d_j + My_j, \quad \forall j = 1, 2, \ldots, n \tag{2.11c}$$

$$t_j \geq 0, \quad \forall j = 1, 2, \ldots, n \tag{2.11d}$$

$$y_j \in \{0, 1\}, \quad \forall j = 1, 2, \ldots, n \tag{2.11e}$$

where y_j is a binary variable taking 1 if task is delayed and 0 otherwise.

2.3.3 Exercises

Exercise 1. The company Tapisall must fulfill a command of three models of wallpaper: one (model 1) has a blue background with green patterns, another (model 2) has a green background with blue patterns, and the last (model 3) is entirely in green. Each model is manufactured as a roll of continuous paper that passes over several machines, each printing a different color (machine 1 for blue color and machine 2 for green). The order of passage on the machines varies from model to model: we first print the blue background on model 1, then the green patterns. For model 2, apply the green background, then the blue patterns. We also know that for operational reasons model 3 must be processed first on machine 2 (compared to the other two models on the same machine) and that any started task must be completed before starting another one. The execution time of these operations is variable according to the surface to be printed. The times (in minutes) required to apply each color to each model are denoted with p_{ij}, that is duration of process i on machine j.

Exercise 2. Assuming that there is no solution satisfying all deadline constraints of one-machine problem, one is looking for minimization of total delays. We assume that all ready times are not an issue as they are fixed at $r_j = 0$ for each job j. Modify the above formulation to handle this objective. Consider also the case when one looks for minimization of the number of delayed tasks.

2.4 Additional exercises

Exercise 1. The cafeteria

To operate a cafeteria, the manager must ensure on-call duty based on the statistics on the required staff. Then number of desired employees on day j is denoted with D_j.

Question: Calculate the minimum number of employees to hire knowing that an employee works 5 days in a row and then has two days off.

Exercise 2. Running a foundry

A foundry receives a specific order for $1,000$ tons of steel. This steel must meet the following characteristics: it must contain at least 0.45% manganese (Mn) while its percentage of silicon (SI) must be between 3.25 and 5. To cast this steel, the foundry has limited quantities of three types of minerals: A, B and C. The contents expressed in (%) per A, B and C are respectively $4, 1$, and 0.6 for Si and $0.45, 0.5$ and 0.4 for Mn.
The process for producing steel is such that direct addition of Mn is possible. This Manganese is available at a price of 8 million euros (ME) per ton. As for the minerals, they cost respectively 21 ME per thousand tons for type A, 25 ME for B and 15 ME for C.
If the foundry plans to sell the steel produced at 0.45 ME per ton, how should it manufacture the 1000 tons requested in order to maximize profit, knowing that the cost of smelting a ton of mineral is 0.005 ME?

Exercise 3. Balanced loading

N SNCF wagons with a payload limited to a weight P are reserved for transporting m boxes. The boxes $1, 2, \ldots, m$ and their weight $p_1, p_2, .., p_m$ are known. It is accepted that there is a way to put all boxes in the wagons.
Question 1. How to allocate the boxes to the wagons so as to respect the maximum payloads and to minimize the load of the most loaded wagon. Model the problem as a linear program.
Question 2. Study the case of two wagons. Make the connection with the (well-known) knapsack problem.

Exercise 4. Backing up files

Before going on vacation you want to make floppy disk backups of important files. You have at your disposal p blank floppy disks with capacities of C GB. They are given the size of the n files that you want to save. Assuming that you do not have any programs to compress the data and that you have enough floppy disks available for back up everything, how to distribute these files on the floppy disks in order to minimize the number of floppy disks used.

Exercise 5. Street surveillance with cameras

A municipality wants to install cameras at the intersections of an industrial zone following repeated thefts. It is assumed that each portion of the street between two intersections is in a straight line. A camera installed at an intersection can rotate 360 degrees and see all the adjacent streets. The municipality wishes to install a minimum number of cameras. Suggest a mathematical model for this problem. The surveillance zone can be represented by a graph where the nodes are the crossroads and the arcs are the streets.

Exercise 6. A transport problem

An Italian transport company must send empty containers from its n depots to m ports. The number of containers available in the depots is denoted with a_i. Container requirements in ports are denoted with b_j. Transport of containers by barges. Each barge can only contain two containers and the cost of transport (per barge) is proportional to the distance traveled (p Euros/km). The distances are given and denoted with D_{ij}.
Formulate as a Linear Program the minimal cost transport problem.

Exercise 7. A transport problem: delivery of heavy packages

A transport company has to deliver to suburban customers n packages with known weights. The transport company has p vehicles that can carry the maximum payloads at the costs also known. Each vehicle can only deliver in the same day packages whose sum of weights does not exceed its payload.
Question 1. Determine the minimum cost transport daily plan (vehicle / package). Give the corresponding Linear Program.
Question 2. How can I be sure that packages l and q will not be delivered by the same vehicle?

Exercise 8. Combined transport

A cargo of T tons must be transported on a route of n cities, with three modes of transport to choose from: rail, road, air. You can change mode at each of the intermediate cities, but the cargo must take a single mode between two consecutive cities. The transport costs in Euros per ton between city pairs are given parameters as well as the costs of mode changes.
Question. Model the problem as a Linear Program whose objective is to minimize the total cost of transport.